CLAIMS

What is claimed is:

1. A method for optimizing the representation of a code sequence, comprising:
determining the frequency of operations performed in the code sequence; and
tuning an instruction set for assigning an op-code representation to an instruction,
wherein the tuning of the instruction set is based on the frequency of operations performed.

2. The method of claim 1, wherein the representation of a code sequence is a bit symbol representation.

3. The method of claim 1, wherein the instruction set is a variable length instruction set.

4. The method of claim 1, wherein the instruction set is a constant length instruction set.

5. The method of claim 1, wherein the step of determining operation frequency may further include loop analysis.

6. The method of claim 1, wherein the modification of the op-code may be executed by a loadable microcode.

7. The method of claim 1, further comprising the step of providing a representation of operation frequency, which represents the frequency of operations performed.

8. The method of claim 7, wherein the representation of operation frequency is a frequency distribution.

9. The method of claim 8, wherein the frequency distribution is a histogram.

10. The method of claim 1, wherein a more compact version of the code sequence is accomplished through shortening of bit symbol representation of an op-code of the instruction set.

11. A method for optimizing the representation of a code sequence, comprising:

determining the frequency of operations performed in the code sequence;

providing a plurality of pre-determined instruction sets with each pre-determined instruction set comprising instructions including assigned op-code representations;

selecting one of the plurality of predetermined instruction sets based on the determined frequency of operations performed.

12. The method of claim 11, wherein the representation of a code sequence is a bit symbol representation.

13. The method of claim 11, wherein the instruction set is a variable length instruction set.

14. The method of claim 11, wherein the instruction set is a constant length instruction set.

15. The method of claim 11, wherein the step of determining operation frequency may further include loop analysis.

16. The method of claim 11, wherein the modification of the op-code may be executed by a loadable microcode.

17. The method of claim 9, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed.

18. The method of claim 17, wherein the representation of operation frequency is a frequency distribution.

19. The method of claim 18, wherein the frequency distribution is a histogram.

20. The method of claim 11, wherein a more compact version of the code sequence is accomplished through shortening of bit symbol representation of an op-code of the instruction set.

21. A method for optimizing the representation of a code sequence, comprising:

determining the frequency of use of a register by the operations performed in the code sequence;

tuning an instruction set for assigning a target-code representation for the register,

wherein the tuning of the instruction set is based on the frequency of use of the register.

22. The method of claim 21, wherein the representation of a code sequence is a bit symbol representation.

23. The method of claim 21, wherein the instruction set is a variable length instruction set.

24. The method of claim 21, wherein the instruction set is a constant length instruction set.

25. The method of claim 21, wherein the step of determining operation frequency may further include loop analysis.

26. The method of claim 21, wherein the modification of the op-code may be executed by a loadable microcode.

27. The method of claim 21, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed.

28. The method of claim 27, wherein the representation of operation frequency is a frequency distribution.

29. The method of claim 28, wherein the frequency distribution is a histogram.

30. A method for optimizing the representation of a code sequence, comprising:

determining the frequency of use of one or more registers within a plurality registers by the operations performed in the code sequence;

limiting the use of one or more of the plurality of registers based on the frequency of use of one or more of the plurality of registers; and

tuning the instruction set for assigning a target-code representation for one or more of the plurality of registers,

wherein the tuning of the instruction set is based on the frequency of use of the plurality of registers.

31. The method of claim 30, wherein the representation of a code sequence is a bit symbol representation.

32. The method of claim 30, wherein the instruction set is a variable length instruction set.

33. The method of claim 30, wherein the instruction set is a constant length instruction set.

34. The method of claim 30, wherein the step of determining operation frequency may further include loop analysis.

35. The method of claim 30, wherein the modification of the op-code may be executed by a loadable microcode.

36. The method of claim 30, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed.

37. The method of claim 36, wherein the representation of operation frequency is a

frequency distribution.

38. The method of claim 37, wherein the frequency distribution is a histogram.

39. A computer readable medium including computer readable code for optimizing a code sequence, comprising:

a first executable code for determining the frequency of operations performed in the code sequence; and

a second executable code for tuning an instruction set for assigning an op-code representation to an instruction,

wherein the tuning of the instruction set is based on the frequency of operations.

40. The computer readable medium of claim 39, wherein the representation of a code sequence is a bit symbol representation.

41. The computer readable medium of claim 39, wherein the instruction set is a variable length instruction set.

42. The computer readable medium of claim 39, wherein the instruction set is a constant length instruction set.

43. The computer readable medium of claim 39, wherein the step of determining operation frequency may further include loop analysis.

44. The computer readable medium of claim 39, wherein the modification of the op-code may be executed by a loadable microcode.

45. The computer readable medium of claim 39, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed.

46. The method of claim 45, wherein the representation of operation frequency is a frequency distribution.

47. The method of claim 46, wherein the frequency distribution is a histogram.

48. An optimized computing assembly, comprising:

a processor coupled with a memory for executing programs;

an optimized code generator operationally coupled with the processor and the memory, the optimized code generator for determining the frequency of operations performed in the code sequence and tuning an instruction set for assigning an op-code representation to an instruction,

wherein the tuning of the instruction set is based on the frequency of operations.

49. The optimized computing assembly of claim 48, wherein the representation of a code sequence is a bit symbol representation.

50. The optimized computing assembly of claim 48, wherein the instruction set is a variable length instruction set.

51. The optimized computing assembly of claim 48, wherein the instruction set is a constant length instruction set.

52. The optimized computing assembly of claim 48, wherein the step of determining operation frequency may further include loop analysis.

53. The optimized computing assembly of claim 48, wherein the modification of the op-code may be executed by a loadable microcode.

54. The optimized computing assembly of claim 48, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed.

55. The method of claim 54, wherein the representation of operation frequency is a frequency distribution.

56. The method of claim 55, wherein the frequency distribution is a histogram.

57. An optimized code generator for a source code, comprising:

a read executable for reading the source code;

a translation executable operationally coupled with the read executable, the translation executable for translating source code to intermediate code

a scanning executable operationally coupled with the translation executable, the scanning executable for determining the frequency of operations performed by the source code and providing a representation of the frequency of operations performed;

an optimizing translation executable operationally coupled with the scanning executable, the optimizing translation executable for translating the intermediate code to object code including an optimized instruction set based on the determined frequency of operations; and

a write executable operationally coupled with the optimizing translation executable, the write executable for outputting the optimized object code.

58. The optimized code generator of claim 57, comprising a compiler.

59. The optimized code generator of claim 57, comprising an assembler.

60. The optimized code generator of claim 57, wherein the instruction set is a variable length instruction set.

61. The optimized code generator of claim 57, wherein the instruction set is a constant length instruction set.

62. The method of claim 57, wherein the representation of operation frequency is a frequency distribution.

63. The method of claim 62, wherein the frequency distribution is a histogram.

64. An optimized code generator, comprising:

means for determining the frequency of operations performed in the code sequence; and

means for tuning an instruction set for assigning an op-code representation to an instruction,

wherein the tuning of the instruction set is based on the frequency of operations.

65. The optimized code generator of claim 64, comprising a compiler.

66. The optimized code generator of claim 64, comprising an assembler.

67. The optimized code generator of claim 64, wherein the instruction set is a variable length instruction set.

68. The optimized code generator of claim 64, wherein the instruction set is a constant length instruction set.

69. The optimized code generator of claim 64, wherein a representation of operations frequency is a frequency distribution.

70. The optimized code generator of claim 69, wherein the frequency distribution is a histogram.